SECTION 1
QUICC/POWERQUICC DIFFERENCES

The following section describes how to move applications from the MC68360 QUICC environment to the MPC860 PowerQUICC environment. It is assumed that the user is familiar with the differences between the 68K type bus used by the QUICC and the PowerPC architechture used by the MPC860. This section is intended to address moving software from one application to the other.

Freescale Semiconductor, Inc.

1.1 CPU AND COMPILERS

The QUICC contains a CPU32+ processor. This executes MC68000 style code and supports instructions implemented in 68K family parts up through the 68020. The PowerQUICC contains the Embedded PowerPC core, which is PowerPC compliant through Book 1. Because moving from a CPU32+ core to an Embedded PowerPC core involves a leap from one architecture to another, existing QUICC code accessing the core must be recompiled. Many tool vendors now have a C cross compiler for 68K to PowerPC instruction set.

Also note that the MPC860 has cache and MMU functionality on chip, which most code will want to take advantage of for performance and code protection reasons.

Major differences between a 68K style design and a PowerPC architechture style design include the fact that the MSbit is bit 0 and the highest interrupt level is 0 not 7. Also note that the nomenclature defines a word as a 32 bit structure and a half word as a 16 bit structure. The bus signals are conceptually similar, but have different names and imply slightly different operation.

1.2 SIU

The SIU on the PowerQUICC is an entirely new design from the SIM on the QUICC. The features of the SIU on the PowerQUICC are a superset of the features supported on the QUICC's SIM, but because of the architechture change in the core, the SIU is a new design.

1.2.1 Memory Controller

The PowerQUICC memory controller is a flexible, easy to use programmable state machine which allows the user complete control over memory cycle accesses. Using DRAM as an example, in the case of the PowerQUICC memory controller, instead of having a set relationship between RAS and CAS, the user may control the RAS/CAS within a resolution of a 1/4 clock cycle. This is performed though a single 32 bit word programmed in internal RAM. Each of the 32-bit words represents the behaviour of the memory control signals for a single

© Freescale Semiconductor, Inc., 2004. All rights reserved.

For More Information On This
Go to: www.freescale

freescale™ semiconductor

system clock period. Bursts on a 32-bit PowerPC bus are 4 beats or a maximum of 8 clock periods. Any memory change on a board, even in the case of changing symmetric memory out for asymmetric memory may be supported by simply reprogramming the Memory controller of the MPC860. The CS0 line is available at boot so that a glueless interface to EEPROM, SRAM and other memory is still supported. The logic portion entitled the User Programmable Machine supports DRAM and DRAM-type memories and the logic block called the General Purpose Chip Selects Machine supports SRAM, EEPROM and other similar memory types. Along with the UPM and the GPCSM are registers specifying base addreses of memory types and the options for accessing each memory attached to the PowerQUICC in the system.

1.2.2 PCMCIA

The PCMCIA socket controller is completely new from the standpoint of the QUICC. Any code written for the QUICC which will take advantage of thePCMCIA functionality of the PowerQUICC should have a PCMCIA driver added to the existing code.

1.2.3 Development Port

The debug interface is completely redesigned between the QUICC and the PowerQUICC due to the architechture change between the CPU32+ and the 505 PowerPC compliant CPU cores. The MPC860 implements a prefetch queue combined with parallel, out of order, pipelined execution, and most fetch cycles are performed internally (e.g. from the I-cache). These features increase performance between the QUICC and PowerQUICC but require new software to handle program trace operations. The Development Port can operate in two working modes: the trap enable mode and debug mode.

1.2.3.0.1 Program Trace. Reconstructing a program trace requires that the program code and some additional information from the MPC860 be available. Reporting on program trace during instruction retirement could significantly complicate debug support, therefore program trace is reported during instruction fetch. Because not all fetched instructions eventually retire (i.e execute normally and are removed from the instruction queue), an indication on canceled instructions is provided to allow the user to reconstruct the actual program flow. When program trace is required, the external hardware should sample the status pins (**VF** & **VFLS**) each clock and also sample the address of all cycles marked with the <inst>Program trace cycle attribute.

1.2.3.0.2 Watchpoints and Breakpoints. The MPC860 CPU supports internal watchpoints (watchpoints that are generated inside the core), internal breakpoints (breakpoints that are generated inside the core), and external breakpoints (breakpoints that are generated outside the core). In general, breakpoints are recognized in the CPU only when the MSR_{RI} bit is set, which guarantees machine restartability after a breakpoint. In the CPU, as in other RISC processors, saving/restoring machine state on the stack during exception handling is done mostly in software. When the software is in the process of saving/restoring machine state, the MSR_{RI} bit is cleared and a breakpoint assertion could cause a non-restartable

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

machine state. In the case where it is necessary to enable breakpoints when the MSR_{RI} bit is clear, both internal breakpoints and an external development system have a *non-maskable* mode.

1.2.4 Clock Generation

The MPC860 Clock Module consists of the main crystal oscillator (OSCM), the System PLL (SPLL), the Low Power Divider, the Clock Generator and Driver blocks and Clock Module & System Low Power Control Block. The Clock Module & System Low Power Control Block receives control bits from the System Clock Control Register (SCCR) and the PLL, Low Power and Reset Control Register (PLPRCR).

Like the QUICC, The MODCK1 and MODCK2 pins are used during Power On Reset (POR) to configure the clock source for the SPLL and the Clocks Drivers. Also, the PLL on the MPC860 can multiply the input frequency by any integer between 1 and 4096. The multiplication factor may be changed by changing the value of MF[0:11] bits in PLPRCR.

The General System Clocks (GCLK1C, GCLK2C, GCLK1 and GCLK2) are the basic clock supplied to all modules and sub-modules on the MPC860. The general system clock can be operated at three frequencies as determined by the DFNL value in SCCR.

The General System clock can switch automatically from “low” to “high” frequency whenever one of the following conditions exists:

- There is a pending interrupt from the Interrupt Controller. This option is maskable by the PRQEN bit in SCCR.
- Power Management (POW) bit in the MSR of the PowerPC Core is cleared (normal operation). This option is maskable by the PRQEN bit in SCCR.
- The CPM RISC controller has a pending request or is currently executing a routine (i.e. it is not idle). This option is maskable by the CRQEN bit in SCCR.

When neither of these conditions exists and the CSRC bit in PLPRCR is set, the general system clock switches automatically back to the low frequency.

The following pins are dedicated to the PLL operation in addition to those on the QUICC: VSSYN, VSSSYN1

1.3 CPM

The CPM on the PowerQUICC is nearly identical to that on the QUICC, with the addition of a few features. The CPMs operate at approximately the same throughput when running at identical frequencies, although a 40MHz PowerQUICC will support four SCCs running ethernet where a QUICC will only run up to 33MHz and supports ethernet on three channels. Each subsection of the CPM is discussed below with the relevent changes described in detail.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

1.3.1 RISC

The PowerQUICC CPM is a superset of the QUICC CPM.

1.3.1.1 MICROCODE. The RISC on the PowerQUICC and QUICC are nearly identical with respect to microcode packages. The PowerQUICC microcode instruction set is a superset of that for the QUICC so backwards compatiblityis maintained. Note that microcode written for the QUICC must be recompiled before being downloaded to the PowerQUICC. There are a few QUICC microcode packages which have been incorporated into the hardware of the PowerQUICC and these microcode packages need not be downloaded to the PowerQUICC. The following list represents these functions

- Asynchronous HDLC
- Signalling System #7

1.3.1.2 DSP. The PowerQUICC provides a basic set of DSP functions in addition to the functions supported on the QUICC. The ROM provides a library of DSP functions based on the Multiply and Accumulator on the CPM which can be combined together to perform various filtering and modulation operations required to support many modem standards including V.32bis and V.34..

1.3.2 Serial Management Controllers (SMC)

The SMCs have not changed between the QUICC and PowerQUICC

1.3.3 Time Slot Assigner

The Time Slot Assigner has not changed between the QUICC and PowerQUICC

1.3.4 Serial Direct Memory Access (SDMA)

Because the SMDA accesses the internal U-bus on the MPC860, the SDMA has added burst capability as well as the ability to burst data in both big and little endian modes.These features are programmed in the receive and transmit function code registers associated with the SCCs, SMCs, SPI and I^2C.

On the MPC860, the cache, SIU, and the SDMA may all become internal bus masters. To determine the relative priority of these masters, each is given an arbitration ID. The 12 SDMA channels share the same ID, which is programmed by the user. Therefore, any SDMA channel can arbitrate for the bus against the other internal masters and any external masters that are present.

Once an SDMA channel obtains the system bus, it remains the bus master for one transaction (which maybe a byte, half-word, word, or burst) before relinquishing the bus. In the case of character-oriented protocols, the SDMA writes characters to memory (it does not wait for multiple characters to be received before writing), but the SDMA always reads long words.

The read or write operation may take multiple bus cycles if the memory provides less than a 32-bit port size. For instance, a 32-bit long-word read from a 16-bit memory will take two SDMA bus cycles. The entire operand (4-word burst, 32 bits on reads and 8, 16, or 32 bits

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

on writes) will be transferred in back-to-back bus cycles before the SDMA relinquishes the bus.

The SDMA can steal cycles with no arbitration overhead when the MPC860 is the bus master.

1.3.5 Independent Direct Memory Access (IDMA)

The MPC860 uses a virtual IDMA rather than a hardware IDMA as in the QUICC. Virtual refers to the fact that the IDMA is implemented via a microcode in ROM on the PowerQUICC. Differences between the QUICC IDMA and the PowerQUICC IDMA relate only to the fact that the PowerQUICC supports both big and little endian modes. Because of this, the IDMA buffer descriptors have an added field specifying the byte ordering of the source and destination operations. The other change relating to the IDMA is the function code for the source operation is within the IDMA buffer descriptor and is no longer in the register memory map as in the QUICC.

Fly-by and double-buffer modes are supported by the virtual IDMA implementation, but note that the RISC microcontroller task scheduling may demonstrate a slight variance in performance. In other words, if the RISC is not performing another task the IDMA operation is faster than that on the QUICC because of the burst capability. However, if the RISC is currently involved in another task and cannot service the IDMA then each IDMA transfer (byte, half-word, word or burst) will not begin until the RISC is free to service it.

1.3.6 Timers

The RISC timers on the MPC860 PowerQUICC are enhanced beyond the functionality provided on the MC68360 QUICC. Each pair of timers may be used to generate a PWM waveform on one of PORT B pins. A maximum of eight channels can be supported. When using this mode of functionality, the first (even numbered) timer is used to control the cycle time of the waveform, the second timer is used to control the duty cycle and the respective Port B pins are configured as a general purpose outputs.

1.3.7 Dual Port RAM

The MPC860 PowerQUICC has 5120 bytes of static RAM configured as dual-port memory used by the CPM on the chip. The dual-port RAM can be accessed by the RISC controller or one of two bus masters: the PowerPC CPU or the SDMA channels.

Within the dual-port RAM the CPM maintains a section of memory called the parameter RAM. This RAM contains many parameters for the operation of the SCCs, SMCs, SPI, I^2C and the IDMA channels.

1.3.8 Parallel I/O Changes

Due to added functionality in the PowerQUICC, a parallel port was added to the CPM. Some of the pins controlled by Port A and B are now controlled by Port D. The following registers have had bit changes between the QUICC and PowerQUICC and software controlling the

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

definitions of these pins should be re-written. PADAT, PADIR, PAODR, PAPAR, PBDAT, PBDIR, PBODR, PBPAR. Pins affected by this change include RXD3, TXD3, RTS3, CTS3, etc.

1.3.9 Interrupt Controller

There is an additional interrupt source from the QUICC to the PowerQUICC. The QUICC had 28 interrupt sources, the PowerQUICC has 29 because of the addition of the I^2C controller. Like the QUICC, the PowerQUICC CPIC supports a fully nested interrupt environment that allows a higher priority interrupt from another CPM source to suspend a lower priority interrupt's service routine. This nesting is achieved by the CPM interrupt in-service register (CPIC). When an interrupt event occurs, the interrupt mask bit in the CPU machine status register (MSR) is cleared to disable further interrupt requests until the s/w is ready to handle them.

It is possible to allow a higher priority interrupt within the same interrupt level to be presented to the CPU core before the servicing of a lower priority interrupt is completed. When the CPU acknowledges an interrupt by setting the IACK bit, the in-service bit for that interrupt source is set in the CISR . This prevents any subsequent CPM interrupt requests at this priority level or lower (within the CPIC interrupt table), until the servicing of the current interrupt has completed and the in-service bit is cleared by the user. Note that pending interrupts for these sources are still set in an interrupt event register during this time, but are set in the CIVR register rather than what was called the CPIC on the QUICC.

The following text provides an example of a handler for an interrupt source with multiple events - in this example SCC1. Note that the bit in CIPR does not need to be cleared by the handler, but the bit in the CISR does need to be cleared.

1. Set the IACK bit in the CIVR
2. Read vector to access interrupt handler.
3. Immediately read the SCC1 event register (SCCE1) into a temporary location.
4. Handle events in the SCC1 Rx or Tx BD tables.
5. Clear the SCC1 bit in the CISR.
6. Execute the RFI instruction. If any unmasked bits in SCCE1 remain at this time (either not cleared by the software or set by the MPC860 during the execution of this handler), this interrupt source will be made pending again immediately following the RFI instruction.

1.4 JTAG

The JTAG interface on the PowerQUICC is completely redesigned from the QUICC. The BSDL file and any tester programs taking advantage of the JTAG interface on the QUICC must be re-written entirely to represent the new scan-chain on the PowerQUICC.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

For More Information On This Product,
Go to: www.freescale.com